

Equivalence Problems for Mappings on Infinite Strings*

KAREL CULIK II AND JAN K. PACHL

*Department of Computer Science, University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada*

This paper is concerned with sets of infinite strings (ω -languages) and mappings between them. The main result is that there is an algorithm for testing the (string by string) equality of two homomorphisms on an ω -regular set of infinite strings. As a corollary we show that it is decidable whether two functional finite-state transducers define the same function on infinite strings (are ω -equivalent).

0. INTRODUCTION

Infinite strings and sets of them (ω -languages) have been extensively studied, see, e.g., Eilenberg (1974) and Cohen and Gold (1977). Finite transducers on infinite strings were considered by Boasson and Nivat (1979).

Our main interest here is testing the equivalence of finite transducers on infinite strings. We consider finite transducers with accepting states (for precise definition see Section 2). It is known that for finite transducers (rational transductions) the equivalence on finite strings is undecidable in general but decidable for functional transducers (Berstel, 1979). The latter result holds even when the domain is restricted to a context free language (Culik, 1979). Our main goal is to prove analogous results for finite transducers working on infinite strings (but we do not consider any domain restriction). The equivalence of two finite transducers on finite strings implies also their ω -equivalence, however, the converse does not hold since two ω -equivalent transducers might produce their outputs at different speeds. Thus ω -equivalence is a necessary condition for equivalence but does not easily reduce to it.

In Section 2 we show some auxiliary results on deterministic ω -regular languages. In the next section we extend the techniques from Salomaa (1978), Culik and Salomaa (1978) and Culik (1979) and show that the (string by string) equivalence of two homomorphisms on an ω -regular set is decidable. Then we show that the ω -equivalence problem for functional

* This research was supported by the Natural Sciences and Engineering Council of Canada, under Grant No. A7403.

transducers reduces to it. Finally, we note that the undecidability of the ω -equivalence for nondeterministic finite transducers (or gsm) easily follows from the undecidability of their equivalence.

1. PRELIMINARIES

Our basic terminology is a mixture of (Eilenberg, 1974) and (Cohen and Gold, 1977).

For a finite alphabet Σ , let Σ^* be the set of finite strings over Σ , Σ^ω the set of infinite strings over Σ , and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. The empty string is denoted by ε , the length of a string w in Σ^* by $|w|$.

We consider two classes of subsets of Σ^ω (ω -languages): ω -regular and deterministic ω -regular ω -languages. We write a (nondeterministic) *finite-state automaton* as $M = (K, \Sigma, \delta, q_0, F)$, where K is a finite set of states, Σ is the input alphabet, $\delta: K \times \Sigma \rightarrow 2^K$ is the transition function, $q_0 \in K$ is the initial state and $F \subseteq K$ is the set of final states. For $\alpha \in \Sigma^\omega$, $\alpha = a_0 a_1 a_2 \dots$, and an infinite sequence $r = p_0 p_1 p_2 \dots$ of states $p_i \in K$, we say that r is a *run of M on α* if $p_0 = q_0$ and $p_i \in \delta(p_{i-1}, a_{i-1})$ for $i \geq 1$. For an infinite sequence of states $r = p_0 p_1 p_2 \dots$, the set of states that appear infinitely many times in r is denoted by $\text{INS}(r)$. The ω -language accepted by automaton M is defined as

$$L^\omega(M) = \{\alpha \in \Sigma^\omega \mid \text{there is a run } r \text{ on } \alpha \text{ such that } \text{INS}(r) \cap F \neq \emptyset\}.$$

In the notation of Cohen and Gold (1977), $L^\omega(M) = T_2(M)$; in the notation of Eilenberg (1974), $L^\omega(M) = \|M\|$.

An ω -language is ω -regular (deterministic ω -regular) if $L = L^\omega(M)$ for some finite-state (deterministic finite state) automaton M . We refer to Eilenberg (1974) and Cohen and Gold (1977) for the basic properties of these two (distinct) classes of ω -languages. A set $L \subseteq \Sigma^\infty$ is ∞ -regular (deterministic ∞ -regular) if $L \cap \Sigma^*$ is a regular language and $L \cap \Sigma^\omega$ is an ω -regular (deterministic ω -regular) ω -language.

A map $g: \Sigma^\infty \rightarrow \Delta^\infty$ is a *proper homomorphism* if $g(\Sigma) \subseteq \Delta^*$, $g: \Sigma^* \rightarrow \Delta^*$ is a homomorphism and $g(a_0 a_1 a_2 \dots) = g(a_0) g(a_1) g(a_2) \dots$ for every ω -word $a_0 a_1 a_2 \dots \in \Sigma^\omega$. For $g: \Sigma^\infty \rightarrow \Delta^\infty$, we write $|g| = \max\{|g(a)| \mid a \in \Sigma\}$.

A proper homomorphism $g: \Sigma^\infty \rightarrow \Delta^\infty$ is *effectively given* by listing $g(a)$ for all $a \in \Sigma$. An ω -regular language L is *effectively given* by giving a finite-state automaton M such that $L = L^\omega(M)$.

When $\alpha, \gamma \in \Sigma^\infty$, we write $\alpha \leq \gamma$ to mean that α is a prefix of γ . When $v \in \Sigma^*$, we denote the string $v v v \dots$ by v^ω .

For basic notions in formal languages see Salomaa (1973).

2. DETERMINISTIC ω -REGULAR ω -LANGUAGES

Here we state three simple results that will be needed later. The first lemma is probably known but we have found no reference.

LEMMA 1. *Deterministic ω -regular ω -languages are effectively closed under intersection.*

Proof. Let $M_i = (K_i, \Sigma, \delta_i, q_{i0}, F_i)$, $i = 1, 2$, be two deterministic finite-state automata. Define $M_3 = (K_3, \Sigma, \delta_3, q_{30}, F_3)$ by

$$\begin{aligned} K_3 &= \{0, 1, 2\} \times K_1 \times K_2, \\ F_3 &= \{2\} \times K_1 \times K_2, \\ q_{30} &= (0, q_{10}, q_{20}), \\ \delta_3((j, q_1, q_2), a) &= (j', q'_1, q'_2) \quad \text{if } q'_i = \delta_i(q_i, a) \\ &\quad \text{for } i = 1, 2 \text{ and } j' = \pi(j, q'_1, q'_2), \end{aligned}$$

where the function $\pi: \{0, 1, 2\} \times K_1 \times K_2 \rightarrow \{0, 1, 2\}$ is defined by

$$\begin{aligned} \pi(2, p, q) &= 0, \\ \pi(0, p, q) &= 0 \quad \text{if } p \notin F_1 \\ &= 1 \quad \text{if } p \in F_1, \\ \pi(1, p, q) &= 1 \quad \text{if } q \notin F_2 \\ &= 2 \quad \text{if } q \in F_2. \end{aligned}$$

From the definition of π it follows that a run of M_3 on $\alpha \in \Sigma^\omega$ enters a state in F_3 infinitely many times if and only if both the corresponding runs of M_1 and M_2 on α enter final states infinitely many times.

Therefore $L^\omega(M_3) = L^\omega(M_1) \cap L^\omega(M_2)$. ■

LEMMA 2. *Deterministic ω -regular ω -languages are effectively closed under union.*

Proof. See Eilenberg (1974). ■

LEMMA 3. *If $g: \Sigma^\omega \rightarrow \Delta^\omega$ is a proper homomorphism then $g^{-1}(\Delta^\omega) \cap \Sigma^\omega$ is a deterministic ω -regular set.*

Proof. $g^{-1}(\Delta^\omega) \cap \Sigma^\omega = L^\omega(M)$ for this finite-state automaton $M = (K, \Sigma, \delta, q_0, F)$: $K = \{q_0, q_1\}$, $F = \{q_0\}$,

$$\begin{aligned} \delta(q_i, a) &= q_1 & \text{if } g(a) = \varepsilon, \quad i = 0, 1; \\ \delta(q_i, a) &= q_0 & \text{if } g(a) \neq \varepsilon, \quad i = 0, 1. \quad \blacksquare \end{aligned}$$

3. ω -EQUALITY SETS

Throughout this section we consider two fixed proper homomorphisms $g, h: \Sigma^\omega \rightarrow \Delta^\omega$; for s in Σ^* we define the *balance* of s by

$$\beta(s) = |g(s)| - |h(s)|,$$

where $|w|$ denotes the length of $w \in \Delta^*$. Let $|\beta| = \max\{|\beta(a)| \mid a \in \Sigma\}$.

The ω -equality set (of g and h) is denoted by E^ω (or $E^\omega(g, h)$ if g, h are not understood) and defined as

$$E^\omega(g, h) = \{\alpha \in \Sigma^\omega \mid g(\alpha) = h(\alpha)\}.$$

For $k = 1, 2, 3, \dots$ let

$$E_k^\omega(g, h) = E_k^\omega = \{\alpha \in E^\omega \mid |\beta(s)| \leq k \text{ for each } s \in \Sigma^*, s \leq \alpha\} \cup$$

$$\bigcup_{0 < |v| \leq k} \{t\gamma \in E^\omega \mid t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \text{ for each } s \leq t, \\ \text{and } [g(\gamma) = v^\omega \text{ or } h(\gamma) = v^\omega]\}.$$

Clearly, $E_1^\omega \subseteq E_2^\omega \subseteq \dots \subseteq E^\omega$.

THEOREM 1. *Each $E_k^\omega(g, h)$ is a deterministic ω -regular ω -language. Moreover, if k, g and h are effectively given then we can effectively give a deterministic finite-state automaton M such that $E_k^\omega(g, h) = L^\omega(M)$.*

Proof. For $v \in \Delta^*$, $0 < |v| \leq k$, let $D(g, h, k, v)$ be the set of the $\alpha \in \Sigma^\omega$ with these two properties:

(1) for each finite $s \leq \alpha$ we have

$$g(s) \leq h(s) \quad \text{or} \quad h(s) \leq g(s);$$

(2) either

(2a) $|\beta(s)| \leq k$ for each finite $s \leq \alpha$

or

$$(2b) \quad \begin{aligned} \alpha = t\gamma, t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \text{ for each} \\ s \leq t, \text{ and } g(s_1) \leq v^\omega, h(ts_1) \leq g(t)v^\omega \text{ for each} \\ \text{finite } s_1 \leq \gamma. \end{aligned}$$

We shall show that each $D(g, h, k, v)$ is (effectively) a deterministic ω -regular ω -language. Then E_k^ω is the union of the sets

$$\begin{aligned} \{ \alpha \in \Sigma^\omega \mid |\beta(s)| \leq k \text{ for each finite } s \leq \alpha \text{ and} \\ g(s) = h(s) \text{ for infinitely many } s \leq \alpha \}, \\ D(g, h, k, v) \cap g^{-1}(\Delta^\omega) \cap h^{-1}(\Delta^\omega), \quad 0 < |v| \leq k, \\ D(h, g, k, v) \cap g^{-1}(\Delta^\omega) \cap h^{-1}(\Delta^\omega), \quad 0 < |v| \leq k. \end{aligned}$$

The first set is deterministic ω -regular (this follows from (Salomaa, 1978, Theorem 2.4); the remaining sets are deterministic ω -regular by Lemmas 1 and 3. Hence the (effective) existence of M follows from Lemma 2.

The deterministic finite state automaton M_1 accepting $D(g, h, k, v)$ works as follows: For an ω -word α on input, M_1 keeps comparing the values of g and h on finite prefixes s of α as long as $|\beta(s)| \leq k$. If and when $|\beta(s)|$ exceeds k , the tails of both $g(\alpha)$ and $h(\alpha)$ are matched against v^ω .

Formally, M_1 has five kinds of states:

- (i) the state $S(+, \varepsilon) = S(-, \varepsilon)$,
- (ii) states $S(+, w)$ and $S(-, w)$ for each $w \in \Delta^*$, $0 < |w| \leq k$,
- (iii) states $S(+, w, u)$ for $w \in \Delta^*$, $0 < |w| \leq k$, $u \leq v$,
- (iv) states $S(u_g, u_h)$ for $u_g \leq v$, $u_h \leq v$.
- (v) the state $S(*)$ (dead state).

All states of M_1 except $S(*)$ are final; the initial state is $S(+, \varepsilon) = S(-, \varepsilon)$.

When running on an infinite string α , the automaton moves through the states $S(\pm, w)$ as long as all finite prefixes p of α encountered satisfy $|\beta(p)| \leq k$; the "buffer" w records the difference between $g(p)$ and $h(p)$ (the sign "+" means $h(p) \leq g(p)$, the sign "-" means $g(p) \leq h(p)$).

If there is a prefix p of α such that $|\beta(p)| > k$, then let t be the longest prefix of α such that $|\beta(s)| \leq k$ for all $s \leq t$. As the automaton reads beyond t , it moves to $S(+, w, u)$ or $S(u_g, u_h)$, depending on whether $h(p) < g(t)$ or $g(t) \leq h(p)$ for the prefix p read so far. The states $S(u_g, u_h)$ trace the progress of $g(p)$ and $h(p)$ through the loop v^ω .

The transition function δ is defined by

$$\begin{aligned}
 \delta(S(+, w), a) &= S(+, w') && \text{if } wg(a) = h(a)w'; \\
 \delta(S(+, w), a) &= S(-, w') && \text{if } wg(a)w' = h(a); \\
 \delta(S(+, w), a) &= S(+, w', u) && \text{if } |wg(a)| > |h(a)| + k, g(a) = v^i u \\
 &&& \text{for some } i \geq 0, w = h(a)w', w' \neq \varepsilon; \\
 \delta(S(+, w), a) &= S(u_g, u_h) && \text{if } ||wg(a)| - |h(a)|| > k, g(a) = v^i u_g \\
 &&& \text{for some } i \geq 0, h(a) = wv^m u_h \text{ for some } m \geq 0; \\
 \delta(S(-, w), a) &= S(+, w') && \text{if } wh(a)w' = g(a); \\
 \delta(S(-, w), a) &= S(-, w') && \text{if } wh(a) = g(a)w'; \\
 \delta(S(-, w), a) &= S(u_g, u_h) && \text{if } ||wh(a)| - |g(a)|| > k, g(a) = v^i u_g \\
 &&& \text{for some } i \geq 0, wh(a) = v^m u_h \text{ for some } m \geq 0; \\
 \delta(S(+, w, u), a) &= S(+, w', u') && \text{if } ug(a) = v^i u' \text{ for some } i \geq 0, \\
 &&& w = h(a)w', w' \neq \varepsilon; \\
 \delta(S(+, w, u), a) &= S(u_g, u_h) && \text{if } ug(a) = v^i u_g \text{ for some } i \geq 0, \\
 &&& h(a) = wv^m u_h \text{ for some } m \geq 0; \\
 \delta(S(u_g, u_h), a) &= S(u'_g, u'_h) && \text{if } u_g g(a) = v^i u'_g \text{ for some } i \geq 0, \\
 &&& u_h h(a) = v^m u'_h \text{ for some } m \geq 0.
 \end{aligned}$$

All other values of δ are $S(*)$.

It follows from the construction that for $x \in \Sigma^*$ we have

$$\begin{aligned}
 \delta(S(+, \varepsilon), x) &= S(+, w) && \text{iff } g(x) = h(x)w \text{ and } |\beta(s)| \leq k \\
 &&& \text{for each } s \leq x; \\
 &= S(-, w) && \text{iff } g(x)w = h(x) \text{ and } |\beta(s)| \leq k \\
 &&& \text{for each } s \leq x; \\
 &= S(+, w, u) && \text{iff there is } t < x, t \text{ is the longest} \\
 &&& \text{prefix of } x \text{ such that } |\beta(s)| \leq k \\
 &&& \text{for each } s \leq t, h(x)w = g(t) \text{ and} \\
 &&& g(x) = g(t)v^i u \text{ for some } i \geq 0; \\
 &= S(u_g, u_h) && \text{iff there is } t < x, t \text{ is the longest} \\
 &&& \text{prefix of } x \text{ such that } |\beta(s)| \leq k \\
 &&& \text{for each } s \leq t, \\
 &&& g(x) = g(t)v^i u_g \text{ for some } i \geq 0, \text{ and} \\
 &&& h(x) = g(t)v^m u_h \text{ for some } m \geq 0; \\
 &= S(*) && \text{otherwise.}
 \end{aligned}$$

Hence $D(g, h, k, v) = L^\omega(M_1)$. ■

Observe that, in the terminology of Cohen and Gold (1977), M also 1'-accepts $D(g, h, k, v)$; hence $D(g, h, k, v)$ is even a regular adherence set in the sense of Boasson and Nivat (1979).

Now, we show a technical lemma which will be needed in the proof of the next theorem.

LEMMA 4. *If $a_1 b_1 c_1 a_2 b_2 c_2 \in \Sigma^*$, $a_1 \leq a_2 b_2$, $a_2 \leq a_1 b_1$, $a_1 b_1^2 c_1 = a_2 b_2^2 c_2$ and $|b_1| = |b_2|$ then $a_1 c_1 = a_2 c_2$.*

Proof. Without loss of generality, assume $a_1 \leq a_2$, i.e., $a_2 = a_1 w$, $w \in \Sigma^*$. Since $a_1 w \leq a_1 b_1$, we have $b_1 = wu$, $u \in \Sigma^*$. Next $a_1 b_1 \leq a_2 b_2$, so $b_2 = uv$, and $|v| = |w|$. Now

$$a_1 w u w u c_1 = a_1 b_1^2 c_1 = a_2 b_2^2 c_2 = a_1 w u v b_2 c_2,$$

hence $v = w$. Since $a_1 w u w u c_1 = a_1 w u w u c_2$, we get $c_1 = w c_2$ and $a_1 c_1 = a_1 w c_2 = a_2 c_2$. ■

Theorem 1 together with the following Theorem 2 enable us to deal with ω -regular subsets of E^ω effectively.

THEOREM 2. *If $R \subseteq \Sigma^\omega$ is ω -regular and $R \subseteq E^\omega(g, h)$ then $R \subseteq E_k^\omega(g, h)$ for some k . More precisely, if $R = L^\omega(M)$ for a finite-state automaton with n states then $R \subseteq E_k^\omega(g, h)$, where $k = n \cdot \max(1, |\beta|, |g|, |h|)$.*

Proof. Take $M = (K, \Sigma, \delta, q_0, F)$ such that $R = L^\omega(M)$; let $k = n \cdot \max(1, |\beta|, |g|, |h|)$, where $n = \text{card}(K)$. Consider any $\alpha \in R$. If $|\beta(s)| \leq k$ for each finite prefix s of α , then $\alpha \in E_k^\omega$; otherwise find the longest finite prefix t of α such that $|\beta(s)| \leq k$ for each $s \leq t$. Write $\alpha = t\gamma$ and let a be the first letter of γ ; clearly $|\beta(ta)| > k$ by the choice of t . We shall assume, without loss of generality, that $\beta(ta) > k$ (the case $\beta(ta) < -k$ is symmetrical), and prove that $g(\gamma) = v^\omega$ for a string v such that $0 < |v| \leq k$. This will prove that $\alpha \in E_k^\omega$.

We have $\beta(t) + |\beta(a)| \geq \beta(ta) > k \geq n \cdot |\beta|$ and $|\beta(a)| \leq |\beta|$, hence $\beta(t) > (n-1) \cdot |\beta|$. Let $r = p_0 p_1 p_2 \dots$ be a run of M on α such that $\text{INS}(r) \cap F \neq \emptyset$; then $p_{|t|} \in \delta(q_0, t)$, and $z\gamma \in R$ whenever $z \in \Sigma^*$ is such that $p_{|t|} \in \delta(q_0, z)$.

Since $\beta(t) > (n-1) \cdot |\beta|$, we can find $u, w, x \in \Sigma^*$ and $q \in K$ such that $|w| \leq n$, $\beta(w) > 0$, $q \in \delta(q_0, u) \cap \delta(q, w)$ and $p_{|t|} \in \delta(q, x)$. It follows that $u w^j x \gamma \in R$, and therefore also $u w^j x \gamma \in E^\omega(g, h)$, for each $j \geq 0$.

We distinguish two cases:

Case I. $h(w) = \varepsilon$. Then $h(u x \gamma) = g(u w^j x \gamma)$ for all j , hence $g(u x \gamma) = g(u w^j x \gamma)$ for all j , hence $g(x \gamma) = g(w)^\omega$. Thus $g(\gamma) = v^\omega$ for some v with $|v| = |g(w)|$. Moreover, $0 < |v| \leq k$ because $0 < \beta(w) = |g(w)| \leq |g| \cdot |w| \leq k$.

Case II. $h(w) \neq \varepsilon$. Since $\beta(w) > 0$, for every finite $s \leq \gamma$ there are $j \geq 0$ and s' such that $g(uw^{2j}) = h(uw^{2j}xs)s'$, $|h(u)| \leq j|h(w)|$, $g(u) \leq h(uw^j)$ and $|g(w)|$ divides j . Thus $j|h(w)| = i|g(w)|$ for some i , and we get $h(uxs) \leq g(uw^{2j})$ by using Lemma 4 with $a_1 = h(u)$, $b_1 = h(w^j)$, $c_1 = h(xs)s'$, $a_2 = g(u)$, $b_2 = g(w^i)$, $c_2 = g(w^{2(j-i)})$.

Therefore $g(ux\gamma) = h(ux\gamma) \leq g(uw^\omega)$, and $g(x\gamma) \leq g(w)^\omega$. Thus either $g(\gamma) = v^\omega$ for some v such that $|v| = |g(w)|$, or $g(\gamma) \in \Delta^*$. We show that the latter is impossible: if $g(\gamma) \in \Delta^*$, then there are $s \in \Sigma^*$ and $\gamma' \in \Sigma^\omega$ such that $\gamma = s\gamma'$, $g(uxs) = h(uxs)$ and $g(\gamma') = h(\gamma') = \varepsilon$. Since $uwx\gamma = uwx s\gamma' \in R$, we also get $g(uwx s) = h(uwx s)$, in contradiction with $|g(uwx s)| = |g(uxs)| + |g(w)| > |h(uxs)| + |h(w)| = |h(uwx s)|$.

This completes the proof. ■

Corollaries to Theorems 1 and 2:

COROLLARY 1. *If the set $E^\omega(g, h)$ is ω -regular then it is deterministic ω -regular.* ■

COROLLARY 2. *Given a finite-state automaton M and two homomorphisms g and h , it is decidable whether or not $g(\alpha) = h(\alpha)$ for each $\alpha \in L^\omega(M)$.*

Proof. By Theorem 2, $g(\alpha) = h(\alpha)$ for each α in $L^\omega(M)$ if and only if $L^\omega(M) \subseteq E_k^\omega$, where $k = n \cdot \max(1, |\beta|, |g|, |h|)$ and n is the number of states in M . By Theorem 1, $E_k^\omega = L^\omega(M_1)$ for some (deterministic) finite-state automaton M_1 . The property $L^\omega(M) \subseteq L^\omega(M_1)$ is effectively testable by (Cohen and Gold, Theorem 2.2.5). ■

4. FINITE-STATE TRANSDUCERS

In this section we look at the infinite behavior of finite-state transducers. We write a (nondeterministic) finite-state transducer as $T = (K, \Sigma, \Delta, \delta, q_0, F)$, where K is a finite set of states, Σ is the input alphabet, Δ is the output alphabet, δ is the transition (and output) function from $K \times (\Sigma \cup \{\varepsilon\})$ into the set of finite subsets of $K \times \Delta^*$, $q_0 \in K$ is the initial state and $F \subseteq K$ is the set of final states. If $\alpha \in \Sigma^\omega$, $\alpha = a_0 a_1 a_2 \dots$, $a_i \in \Sigma \cup \{\varepsilon\}$ for $i \geq 0$, $\gamma \in \Delta^\omega$, $\gamma = u_0 u_1 u_2 \dots$, $u_i \in \Delta^*$ for $i \geq 0$, and $r = p_0 p_1 p_2 \dots$ is an infinite sequence of states $p_i \in K$, we say that r is a *run of T on α with output γ* if $p_0 = q_0$ and $(p_i, u_{i-1}) \in \delta(p_{i-1}, a_{i-1})$ for $i \geq 1$; in symbols we write $\alpha \rightarrow^r \gamma$.

Now we will introduce two relations $R^\omega(T)$ and $R^\omega(T)$ defined by T :

$$R^\omega(T) = \{(\alpha, \gamma) \in \Sigma^\omega \times \Delta^\omega \mid \alpha \xrightarrow{r} \gamma \text{ for some run } r \text{ such that} \\ \text{INS}(r) \cap F \neq \emptyset\},$$

$$R^\omega(T) = R^\omega(T) \cap (\Sigma^\omega \times \Delta^\omega).$$

The relation $R^\infty(T)$ describes the behavior of transducer T on both finite and infinite input strings, the relation $R^\omega(T)$ on infinite input strings only. Note, however, that the range of $R^\omega(T)$ might contain finite strings, because T might read infinite input and produce empty output.

The analogue of Nivat's factorization theorem (Eilenberg 1974, Theorem IX.2.2) takes this form:

THEOREM 3. *For a set $A \subseteq \Sigma^\infty \times \Delta^\infty$, two conditions are equivalent:*

- (i) $A = R^\infty(T)$ for a finite-state transducer T ;
- (ii) *there is an ω -regular set $B \subseteq \Gamma^\omega$ and two proper homomorphisms $g: \Gamma^\infty \rightarrow \Sigma^\infty$ and $h: \Gamma^\infty \rightarrow \Delta^\infty$ such that*

$$A = \{(g(\gamma), h(\gamma)) \mid \gamma \in B\}. \quad \blacksquare$$

The following lemma is easy to verify.

LEMMA 5. *Let $g: \Sigma^\infty \rightarrow \Delta^\infty$ be a proper homomorphism. If $R \subseteq \Sigma^\infty$ is ∞ -regular then $g(R) \subseteq \Delta^\infty$ is ∞ -regular. If $R \subseteq \Delta^\infty$ is ∞ -regular then $g^{-1}(R) \subseteq \Sigma^\infty$ is ∞ -regular.* \blacksquare

THEOREM 4. *If T is a finite-state transducer, then the domain and the range of $R^\infty(T)$ are ∞ -regular sets.*

Proof. Follows by Theorem 3 and Lemma 5. \blacksquare

Moreover, the domain and the range of $R^\infty(T)$ can be effectively given when T is effectively given (because the constructions in Theorem 3 and Lemma 5 are effective); this fact will be used in the proof of Theorem 5.

A finite-state transducer T is called *functional* if the relation $R^\infty(T) \subseteq \Sigma^\infty \times \Delta^\infty$ is a partial function from Σ^∞ to Δ^∞ .

THEOREM 5. *There is an algorithm to decide, for two given functional finite-state transducers T_1 and T_2 , whether $R^\infty(T_1) \subseteq R^\infty(T_2)$.*

Proof. Let $T_i = (K_i, \Sigma, \Delta, \delta_i, q_{i0}, F_i)$, $i = 1, 2$. Let $\bar{\Delta}$ be another copy of Δ , disjoint from Δ ; the correspondence between Δ^* and $(\bar{\Delta})^*$ will be written as $x \mapsto \bar{x}$, $x \in \Delta^*$. Define two proper homomorphisms $g, h: (\Delta \cup \bar{\Delta})^\infty \rightarrow \Delta^\infty$ by $g(a) = a$, $g(\bar{a}) = h(a) = \varepsilon$ and $h(\bar{a}) = a$ for $a \in \Delta$.

We shall construct a finite-state transducer T_3 such that $R^\infty(T_1) \subseteq R^\infty(T_2)$ if and only if

- (i) the domain of $R^\infty(T_1)$ is contained in the domain of $R^\infty(T_2)$, and
- (ii) $g(a) = h(a)$ for each a in the range of $R^\infty(T_3)$.

This together with Corollary 2 (in Section 3) and Theorem 4 establishes the result.

The transducer T_3 simulates the simultaneous operation of T_1 and T_2 and produces their outputs intermixed (the outputs are then separated by g and h). On input $a \in \Sigma$, T_3 outputs $x_1 \bar{x}_2$, where x_1 is the output of T_1 and x_2 is the output of T_2 . The situation is more complicated with input ε : here it can happen that only T_1 or only T_2 moves (this possibility is taken care of by the function δ_5 defined below).

Formally, we define $T_3 = (K_3, \Sigma, \Delta \cup \bar{\Delta}, \delta_3, q_{30}, F_3)$, where

$$K_3 = \{0, 1, 2\} \times K_1 \times K_2,$$

$$q_{30} = (0, q_{10}, q_{20}),$$

$$F_3 = \{2\} \times K_1 \times K_2.$$

Let $\delta_3(p, a) = \delta_4(p, a) \cup \delta_5(p, a)$ for $p \in K_3$ and $a \in \Sigma \cup \{\varepsilon\}$, where δ_4 and δ_5 are defined as follows (the formula for δ_4 employs the function $\pi: \{0, 1, 2\} \times K_1 \times K_2 \rightarrow \{0, 1, 2\}$ defined in the proof of Lemma 1):

$$\begin{aligned} \delta_4((j, q_1, q_2), a) = \{ & ((\pi(j, q'_1, q'_2), q'_1, q'_2), x_1 \bar{x}_2) \mid \\ & (q'_i, x_i) \in \delta_i(q_i, a) \text{ for } i = 1, 2, \\ & a \in \Sigma \cup \{\varepsilon\}; \end{aligned}$$

$$\delta_5((j, q_1, q_2), a) = \emptyset \quad \text{for } a \in \Sigma;$$

$$\begin{aligned} \delta_5((0, q_1, q_2), \varepsilon) = \{ & ((j, q'_1, q_2), x_1) \mid \\ & (q'_1, x_1) \in \delta_1(q_1, \varepsilon) \text{ and either} \\ & [q'_1 \notin F_1 \text{ and } j = 0] \text{ or } [q'_1 \in F_1 \text{ and } j = 1] \} \\ & \cup \{((0, q_1, q'_2), \bar{x}_2) \mid (q'_2, x_2) \in \delta_2(q_2, \varepsilon)\}; \end{aligned}$$

$$\begin{aligned} \delta_5((1, q_1, q_2), \varepsilon) = \{ & ((1, q'_1, q_2), x_1) \mid (q'_1, x_1) \in \delta_1(q_1, \varepsilon)\} \\ & \cup \{((j, q_1, q'_2), \bar{x}_2) \mid (q'_2, x_2) \in \delta_2(q_2, \varepsilon) \\ & \text{and either } [q'_2 \notin F_2 \text{ and } j = 1] \text{ or} \\ & [q'_2 \in F_2 \text{ and } j = 2]\}; \end{aligned}$$

$$\begin{aligned} \delta_5((2, q_1, q_2), \varepsilon) = \{ & ((0, q'_1, q_2), x_1) \mid (q'_1, x_1) \in \delta_1(q_1, \varepsilon)\} \\ & \cup \{((0, q_1, q'_2), \bar{x}_2) \mid (q'_2, x_2) \in \delta_2(q_2, \varepsilon)\}. \end{aligned}$$

The definition of δ_3 and F_3 ensures that if $r = (j_0, q_{10}, q_{20})(j_1, q_{11}, q_{21})(j_2, q_{12}, q_{22})\dots$ is a run of T_3 on $\alpha \in \Sigma^\infty$ and r_1 and r_2 are the corresponding runs of T_1 and T_2 on α , then $\text{INS}(r) \cap F_3 \neq \emptyset$ iff $\text{INS}(r_1) \cap F_1 \neq \emptyset$ and $\text{INS}(r_2) \cap F_2 \neq \emptyset$. Therefore T_3 has the desired properties and the proof is completed. ■

COROLLARY 3. *Given two functional finite transducers T_1 and T_2 , it is decidable whether $R^\infty(T_1) = R^\infty(T_2)$.*

Proof. The decidability of inclusion implies the decidability of equality. ■

We say that two finite transducers T_1 and T_2 are ω -equivalent if $R^\omega(T_1) = R^\omega(T_2)$.

COROLLARY 4. *The ω -equivalence problem for functional finite transducers is decidable.*

Proof. Follows from Corollary 3 and the following lemma, which says that finite transducers on infinite string are closed under the restriction to an ω -regular set. ■

LEMMA 6. *Let $T = (K, \Sigma, \Delta, \delta, q_0, F)$ be a finite transducer and $L \subseteq \Sigma^\omega$ an ω -regular set. Then we can construct a finite transducer T_1 such that $R^\omega(T_1) = R^\omega(T) \cap (L \times \Delta^\omega)$.*

Proof. As in Theorem 3, find $\Gamma, g: \Gamma^\infty \rightarrow \Sigma^\infty, h: \Gamma^\infty \rightarrow \Delta^\infty$ and $B \subseteq \Gamma^\omega$ such that $R^\omega(T) = \{(g(\gamma), h(\gamma)) \mid \gamma \in B\}$. The set $g^{-1}(L) \subseteq \Gamma^\infty$ is ∞ -regular by Lemma 5, and $B \cap g^{-1}(L) \subseteq \Gamma^\omega$ is ω -regular because ω -regular sets are closed under intersection (Eilenberg, 1974, Chap. XIV). Since $R^\omega(T) \cap (L \times \Delta^\omega) = \{(g(\gamma), h(\gamma)) \mid \gamma \in B \cap g^{-1}(L)\}$, the result follows by Theorem 3. ■

We conclude this section by showing that the ω -equivalence is undecidable for general finite transducers, even for gsm as defined in (Salomaa, 1973).

THEOREM 6. *The ω -equivalence problem for (nondeterministic) gsm is undecidable.*

Proof. Consider any gsm $M = (K, \Sigma, \Delta, \delta, q_0, F)$. Modify it to M' so that $M' = (K \cup \{f\}, \Sigma \cup \{\#\}, \Delta \cup \{\#\}, \delta', q_0, \{f\})$, where $f \notin K, \# \notin \Sigma \cup \Delta$, and δ is extended to δ' by $(f, \#) \in \delta'(q, \#)$ for each $q \in F, (f, \#) \in \delta'(f, \#)$.

Clearly M_1 and M_2 are equivalent iff M'_1 and M'_2 , constructed as above, are ω -equivalent. That completes the proof since the equivalence for (nondeterministic) gsm is undecidable (Berstel, 1979). ■

RECEIVED: September 27, 1980; REVISED: April 17, 1981

REFERENCES

- BERSTEL, J. (1979), "Transductions and Context-Free Languages," Teubner, Stuttgart.
 BOASSON, L. AND NIVAT, M. (1979), "Adherences of Languages," Université Paris, Laboratoire Informatique Théorique et Programmation, Res. Rep. N° 79-13.

- COHEN, R. S. AND GOLD, A. Y. (1977), Theory of ω -languages. I. Characterization of ω -context-free languages, *J. Comput. System Sci.* **15**, 169–184.
- EILENBERG, S. (1974). "Automata, Languages and Machines," Vol. A, Academic Press, New York.
- CULIK, K. II (1979), Some decidability results about regular and pushdown translations, *Inform. Process. Lett.* **8**, 5–8.
- CULIK, K. II AND SALOMAA, A. (1978), On the decidability of homomorphism equivalence for languages, *J. Comput. System Sci.* **17**, 163–175.
- SALOMAA, A. (1973), "Formal Languages," Academic Press, New York.
- SALOMAA, A. (1978), Equality sets for homomorphisms of free monoids, *Acta Cybernet.* **4**, 127–139.